延伸阅读

KERNIHAN B W，RITCHIE D M. C 程序设计语言[M]. 徐宝文，李志，译. 2 版. 北京：机械工业出版社，2004：119-120.

（金靖）

1.2.12 文件及基本读写

1.2.12.1 文件的基本概念、文本文件的基本操作

文件指存储在外部介质上数据的集合，简单来讲就是把数据通过字节序列保存在磁盘上。在磁盘上存储、调用数据都是通过文件操作的。根据编码方式，文件可分为两种类型：文本文件和二进制文件。

文本文件的基本操作有读取和写入等。

（金靖）

1.2.12.2 文本文件类型与二进制文件类型

文本文件是一个字符文件。它是基于字符编码的文件，常见的编码有 ASCII 编码、UNICODE 编码等。二进制文件是用计算机的内部格式存储数据集合。二进制文件中的数据只有被程序正确地解释时才有意义，因此要事先知道它的编码形式才能正确解码转换。

（金靖）

1.2.12.3 文件重定向、文件读写等操作

C++程序读写文件的方式有两种：流式和 I/O 方式。信息学竞赛中一般使用流式文件操作，分为两种：文件指针 FILE 和 stream 类的文件流。

1. 输入与输出流

流是与磁盘或其他外围设备关联的数据的源或目的地，C++中提供了文本流和二进制流两种类型，其中文本流是由文本行组成的序列，每一行包含 0 个或多个字符，并以‘\n’结尾。程序开始执行时，stdin、stdout 和 stderr 这 3 个流已经处于打开状态，分别为标准输入、标准输出和标准错误。在大多数环境中，stdin 指向键盘，而 stdout 和 stderr 指向显示器。

2. 文件指针 FILE

FILE 类型结构体用来定义文件指针变量，即文件流，其中封装了与文件有关的信息，如缓冲区、位置指针等。使用时要包含头文件 cstdio。

```
FILE *fopen(const char *filename,const char *mode)
```

fopen 函数用于打开 filename 指定的文件，并返回一个指向它的流。如果打开操作

失败，则返回 NULL。mode 为打开模式，可以为下列合法值之一：

- “r”，以只读方式打开文本文件，若文件不存在则操作失败；
- “w”，以只写方式打开文本文件，并删除已存在的内容，若文件不存在则创建该文件；
- “a”，以只写方式打开文本文件，并向文件末尾追加内容，若文件不存在则创建该文件；
- “+”，添加上面的字符串之后，表示以读写的方式打开文件(既可以读又可以写)；
- “b”，添加上面的字符串之后，表示对二进制文件进行操作。

在对同一文件进行读和写的交叉过程中，必须调用 fflush 函数或文件定位函数。

```
int fflush(FILE *stream)
```

对于输出流来说，fflush 函数将已写入缓冲区但尚未写入文件的所有数据写入文件中。对输入流来说，fflush 函数的结果是未定义的，如果在写的过程中发生错误，则返回 EOF(其值一般为-1)，否则返回 0。

```
int fclose(FILE *stream)
```

fclose 函数将所有未写入的数据写入 stream 中，丢弃缓冲区中的所有未读输入数据，并释放自动分配的全部缓冲区，最后关闭流。若发生错误则返回 EOF，否则返回 0。

3. 格式化文件输入输出

```
int fscanf(FILE *stream,const char *format,…)
```

fscanf 函数按照 format 说明的格式从 stream 流中读取输入，并把转换后的值赋给后续各个参数。当格式串 format 用完时，函数返回。如果到达文件的末尾或在转换输入前发生错误，该函数返回 EOF；否则，返回实际被转换并赋值的输入项的数目。

```
int fprintf(FILE *stream,const char *format,…)
```

fprintf 函数按照 format 说明的格式对输出进行转换，并写入 stream 流中。返回值是实际写入的字符数，若出错则返回一个负值。

fscanf、fprintf 函数中的 format 格式串与 scanf、printf 函数遵循相同的模式。特别指出，fscanf(stdin,…)函数等价于 scanf(…)，fprintf(stdout,…)函数等价于 printf(…)。

4. fstream 类文件流

C++还提供了 fstream 类用于读写文件，使用时要包含头文件 fstream，用于从文件中读写数据。ifstream 用于从文件中读取数据，ofstream 用于向文件中写入数据。

5. 文件重定向

信息学竞赛中通常只需要同时打开一个输入文件和一个输出文件，因此可以使用freopen函数实现输入输出文件重定向，将stdin、stdout等已打开的文件流重定向到指定文件。

```
FILE *freopen(const char *filename,const char *mode,FILE *stream)
```

freopen函数将stream流重新定向到filename指定的文件。mode为打开模式，可以为下列合法值之一：

- “r”，以只读方式打开文本文件，若文件不存在则操作失败；
- “w”，以只写方式打开文本文件，并删除已存在的内容，若文件不存在则创建该文件。

freopen若操作失败则返回NULL，否则返回参数stream。

延伸阅读

KERNIHAN B W，RITCHIE D M. C程序设计语言[M]. 徐宝文，李志，译. 2版. 北京：机械工业出版社，2004：140-141，220.

（金靖）

1.2.13 STL模板

1.2.13.1 算法模板库中的函数：min、max、swap、sort

C++的algorithm库提供了以下几种函数。

1. min函数

`const T& min(const T& a,const T& b)`，利用类型T的<运算符，如果$a<b$则返回a，否则返回b。

2. max函数

`const T& max(const T& a,const T& b)`，利用类型T的<运算符，如果$a<b$则返回b，否则返回a。

3. swap函数

`void swap(T& a,T& b)`，交换a和b这两个地址中的值。如果交换的是两个数组，那么会进行$O(n)$次交换，其中n是数组长度。

4. sort函数

`void sort(*begin,*end,cmp)`，将一段连续的区间[first,last)中的元素排序。cmp是一个bool比较函数，可以省略。当省略时，sort使用<运算符作为比较函数，可以通过重载operator<运算符对复杂类型排序，也可以编写自定义函数cmp来明确排序规则，还可以在如表1.5所示的选项中选择。

表 1.5　sort 函数选项

选项	说明
equal_to	相等
not_equal_to	不相等
less	小于
greater	大于
less_equal	小于等于
greater_equal	大于等于

如果将一个区间的整数按照降序排序，可以写成：`sort(*begin,*end,greater<int>());`。

sort 函数采用的不是简单的快速排序。当数据量大时，sort 函数采用快速排序算法，分段归并排序；一旦分段后的数据量小于某个门槛，为避免快速排序的递归调用带来过大的额外负荷，就改用插入排序；如果递归层次过深，还会改用堆排序(HeapSort)。因此它具有很好的平均性能，时间复杂度为 $O(n\log n)$。

参考词条

1. 排序算法
2. 算法模板库中的常用函数

延伸阅读

BJARNE S. C++程序设计语言(第 4 部分：标准库)[M]. 王刚，杨巨峰，译. 4 版. 北京：机械工业出版社，2016：71-73.

(金靖)

1.2.13.2　栈、队列、链表、向量等容器

C++的 STL 标准模板库中提供了以下几种容器。

1. 栈

栈(stack)是限定仅在表尾进行插入或删除操作的线性表。它按照后进先出的原则存储数据，先进入的数据被压入栈底，最后进入的数据在栈顶。容器定义在头文件 stack 中。对于 stack 类型的变量 stk，基本操作有如表 1.6 所示的几种。

表 1.6　栈的基本操作

方法	功能
stk.empty()	判断栈是否为空
stk.size()	返回栈中元素个数
stk.top()	返回栈顶元素的引用
stk.push(x)	在栈顶加入一个元素
stk.pop()	删除栈顶元素(至少保证有一个元素)

栈操作示例代码如下。

```
stack<int> stk;
for(int i=0;i<5;i++)
    stk.push(i* i);
while(!stk.empty()){
    cout<<stk.top()<<" "<<stk.size()<<endl;
    stk.pop();
}
/*
运行结果:
16 5
9 4
4 3
1 2
0 1
*/
```

2. 队列

队列(queue)是限定在前端(称为队头)进行删除操作，在后端(称为队尾)进行插入操作的线性表。它按照先进先出的原则存储数据。容器定义在头文件 queue 中。对于 queue 类型的变量 q，基本操作有如表 1.7 所示的几种。

表 1.7　队列的基本操作

方法	功能
q. empty( )	返回队列中元素个数是否为 0
q. size( )	返回队列中的元素个数
q. front( )	返回队列中队头元素的引用
q. back( )	返回队列中队尾元素的引用
q. pop( )	删除队头元素
q. push(x)	在队尾插入一个元素

3. 链表

链表(list)是支持常数时间从容器任何位置插入和移除元素的线性表，通常实现为双向链表。容器定义在头文件 list 中。对于 list 类型的变量 lst，基本操作有如表 1.8 所示的几种。

表 1.8　链表的基本操作

方法	功能
lst. empty( )	返回链表中元素个数是否为 0
lst. size( )	返回链表中的元素个数，时间复杂度为 $O(n)$
lst. front( )	返回链表中第一个元素的引用